ELECTRONIC SYSTEMS DIVISION
AIR FORCE SYSTEMS COMMAND
HANSCOM AIR FORCE BASE, MASSACHUSETTS

MCI-76-2

January 1977

ESD 1976 COMPUTER SECURITY DEVELOPMENTS SUMMARY

DIRECTORATE OF COMPUTER SYSTEMS ENGINEERING

DEPUTY FOR COMMAND AND MANAGEMENT SYSTEMS

LEGAL NOTICE

When U.S. Government drawings, specifications or other data are used for any purpose other than a definitely related government procurement operation, the government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

OTHER NOTICES

Do not return this copy. Retain or destroy.

REVIEW AND APPROVAL

This technical report has been reviewed and is approved for publication.

John T Holland
JOHN T. HOLLAND, Lt Col, USAF
Chief, Techniques Engineering
Division

Roger R. Schell
ROGER R. SCHELL, Lt Col, USAF
ADP System Security Program
Manager

FOR THE COMMANDER

Frank J. Emma
FRANK J. EMMA, Colonel, USAF
Director, Computer Systems Engineering
Deputy for Command & Management Systems

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER MCI-76-2	2. GOVT ACCESSION NO. None	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) ESD 1976 COMPUTER SECURITY DEVELOPMENT SUMMARY		5. TYPE OF REPORT & PERIOD COVERED Interim Report
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Richard D. Rhode, MITRE Corporation		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Deputy for Command & Management Systems (MCI) Hq Electronic Systems Division (AFSC) Hanscom AFB, Bedford, MA 01731		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS PE 64740F/Project 2239
11. CONTROLLING OFFICE NAME AND ADDRESS See Item 9		12. REPORT DATE January 1977
		13. NUMBER OF PAGES 42
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

The material presented was originally prepared by Mr. Richard D. Rhode of the MITRE Corporation, Bedford, Massachusetts, in support of Project 5720 under Contract No. F19628-76-C-0001. This document does not quality as an ESD Technical Report, and it is NOT AVAILABLE THROUGH DDC.

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Secure Computer Systems
Multilevel Systems

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This document presents a summary of ESD computer security development efforts that were underway or proposed before Hq AFSC directed that the project be terminated. The purpose is to portray the nature of the relevant problems and viable technical approaches for their solution. The efforts described do not necessarily reflect approved Air Force development projects.

EXECUTIVE SUMMARY

The Air Force multilevel computer security problem arises when a data processing system must store or process multiple levels or compartments of information, and must support users with several levels of clearance. Systems that raise this problem are needed to meet both operational and economic requirements. Operational requirements are encountered when a system cannot be operated by clearing all users for all information -- for example, in an automated interface between intelligence and operations communities. Economic requirements are raised in those environments where a single computer must support processing at a variety of levels -- and where the cost of changing the computer from level to level is unacceptable.

ESD's computer security program was initiated in 1970 in response to an Air Force Data Services Center (AFDSC) requirement to operate Honeywell 635's in a multilevel mode. ESD rapidly discovered that security flaws were pervasive in the GCOS operating system for the 635's, and that any single flaw could permit repeated and undetected access to any stored information by a hostile programmer. ESD could devise no way to assure that all potentially damaging flaws had been found and repaired, but did find that new flaws could be found (even after significant repair effort) with surprising ease. Thus the AFDSC computer security requirement remained unsatisfied.

In a renewed attempt to meet the requirement, ESD convened a computer security technology panel in 1972. The panel suggested that a secure system could be achieved by developing a mechanism that would implement a reference monitor. A reference monitor mediates the accesses of subjects (that act on behalf of users) to objects containing information in a computer system. The "security kernel" that implements the reference monitor must be invoked on every access by a subject to an object, be tamperproof, and be verified to enforce a security policy. The panel also suggested a formal mathematical approach to verifying the correctness of a kernel.

ESD initiated efforts to follow the panel's suggested approach in late 1972. A mathematical model was developed in 1973 that formally describes the operation of a security kernel. A kernel design approach has also been developed, involving the preparation of successively more primitive formal specifications of a kernel's design. Formal proofs link model, specifications, and kernel programs and hardware.

In 1974, The MITRE Corporation developed prototype kernel software to operate on the DEC PDP-11/45 minicomputer hardware. The kernel specifications have been proven secure with respect to the model, and program proofs were underway at the end of 1975, with completion planned for mid-1976. The kernel has been used in a demonstration of application of a secure computer system in an operations- intelligence interface. A file management system and air track display software were developed to support the demonstration. Current work with the PDP-11/45 kernel is centered on interfacing it with the Bell Laboratories UNIX operating system to provide a more efficient and general secure system prototype.

In 1973, in response to AFDSC's specific requirements, ESD initiated the acquisition and development of a Honeywell Multics computer and security enhancements. The enhancements, based on the security model, were completed in late 1975 and installed at AFDSC. While the enhancements do not provide the security of a kernel, they do result in a system adequate for a controlled multilevel environment. AFDSC plans to use the enhanced Multics to process data up to Top Secret in support of timesharing users with Secret and Top Secret clearances.

The development of a version of Multics with a kernel was started in 1974. The development contract with Honeywell requires the preparation and verification of formal specifications and kernel programs. The system produced by this long-term effort is expected to be compatible with user programs and procedures prepared for the current AFDSC Multics, though the security of the system should be adequate to allow uncleared users to be supported. Part of the secure Multics project is the development of a minicomputer front-end processor with its own kernel. This processor, in addition to serving as a communications front-end for the secure Multics, can support general secure minicomputer applications with high efficiency. The secure minicomputer hardware is also to be available in militarized form.

In summary, the ESD computer security program has made significant progress and developed a number of interim products. The program has developed the technology needed to produce secure multilevel computer systems, and has demonstrated the feasibility of that technology. It was well on the way to demonstrating the application of the technology in an operational, certifiable system before Hq AFSC directed that the multilevel security program be terminated during FY77. This paper documents the nature of that program prior to its termination.

SECTION I

INTRODUCTION

This document describes a program to provide Air Force ADP users with the ability to process classified information securely and economically in computer systems. Such an ability is lacking in today´s systems. As a result, procedural "fixes" have been necessarily generated; these fixes have significant costs and have failed to address major operational requirements.

The document begins with an overview of the technical problem of computer security and of the Air Force user requirements that make this problem important. It then outlines a unified technical approach to solving computer security problems, and goes on to summarize major ESD-sponsored developments that use this approach. The Appendix presents a breakdown of the individual tasks that make up the development program.

SECTION II

COMPUTER SECURITY PROBLEMS AND REQUIREMENTS

CURRENT ADP SECURITY PRACTICE

The problem of multilevel security in Automatic Data Processing (ADP) can best be introduced in terms of the special procedures used for processing several levels of classified information. In current ADP systems, two alternatives are generally employed:

> All security levels may be processed together -- provided that all users (and terminal areas and communications) are cleared for the highest level of information that could be processed on the system.
>
> Each level may be processed at a separate time, in which case the entire system environment (terminals, disk packs, tapes, printer ribbons) must be changed or sanitized at each change of security level.

The first alternative produces a proliferation of personnel clearances, secure terminal areas, and secure communications. The second, called "color changing", does not. Even an uncleared terminal may be served provided it is detached before classified processing begins. But each change of level wastes a significant amount of system time while the change of environments is being completed. [1] Regardless which alternative is employed, the procedures necessary today to process multiple levels of classified information with computer systems involve increased cost, inconvenience, and/or system inefficiency.

COMPUTER SECURITY REQUIREMENTS

This subsection summarizes the computer security requirements of some major Air Force ADP users. While it is not exhaustive, it does indicate the major problems that have been encountered with the use of

[1] The terms "security level" and "level of information" are used here to designate a single National Defense Security classification level (Confidential, Secret, etc.) and one set of compartments (formal need-to-know classes).

current non-technological alternatives. Trends in future problems and requirements can be inferred from these experiences. The impacts of computer security requirements on system costs and on operational capabilities are stressed.

It should be noted here that hostile penetrations directed against computers processing classified data are not known to have occurred. However, this is not because such penetrations are impossible, but because current policies dictate operation in the modes described above, precluding such penetrations. Recent policy modifications have offered Air Force ADP managers the option of weakening these restrictions, but most installations have declined to implement the modifications, believing them inconsistent with their responsibilities for protecting classified information.

Cost Impacts

The cost impacts of computer security have been reflected in expenditures for increased protection and additional equipment, and in inefficient system utilization. Typical of the installations that have required increased protection is the Air Force Data Services Center at the Pentagon. There, additional personnel clearances, vault areas, and secure communications have been required to allow users to do unclassified processing on computers that handle classified data. The cost of securing each remote site (excluding terminal equipment) is estimated by AFDSC at $50,000. At the Strategic Air Command, additional SIOP clearances and area protection were required when it was decided that the 4000th Aerospace Applications Group was to receive its computer support from the SAC World Wide Military Command Control System (WWMCCS) ADPE.

Computer installations that must provide responsive support to user communities of varied clearance levels have had to purchase additional equipment. At AFDSC, a timesharing system (a Honeywell 635) was acquired to provide unclassified computing services to AFDSC´s users in open office areas, supplementing the classified processing systems (with secure remote terminals) mentioned above. One of the two SAC WWMCCS dual processors was split into two single processor systems so that development, on-line support and planning applications, each of different security level, could each have its own computer. An additional Honeywell 6080 WWMCCS processor has been installed at MAC to satisfy MAC´s need to provide timely support to classified crisis management applications. The added equipment cost approximately $4 million (an estimated $2 million for the 635 mentioned and $1 million each for the dual processor split and additional 6080). Additional Air Force WWMCCS (and other) computer facilities can be expected to require similar additions of equipment as major classified processing applications become operational.

Inefficient equipment utilization is reflected in the phenomenon of classified processing systems known as the "color change". In a color change, all work of one security level is completed, print queues are drained, and media dismounted. Then system memories are cleared, new media (including the operating system residence) are mounted, and a version of the system is brought up to process the new level. The actual time required to perform the change of media and clear and restart the system ranges from twenty to forty-five minutes. The color change's effect may be propagated over one to two hour's processing by refusing long jobs and by saving files on backup tapes. Color changes are used in cases where responsiveness and workload do not require dedication of a computer to a given level. Thus SAC, with its many WWMCCS computers, performs several color changes each day. MAC and the SAC intelligence computer (a 360/85) also perform color changes, and so do smaller Air Force WWMCCS installations. These changes can easily absorb ten to twenty percent of a system's processing capacity.[2]

Operational Impacts

Where possible, operational requirements for secure computers are met either by adding equipment so that there is a computer for each required level, or by clearing all users for access to all information processed. There is, however, a significant class of operational requirements that cannot be satisfied by today's computer systems using these alternatives.

For example, during the 1973 Middle East War, the Military Airlift Command was required to transport military supplies and equipment into Israel. Because of the sensitive nature of the operation, its details were classified Secret. Because of the operation's classification, it was impossible to support, at the same time with available equipment, both operation of normal unclassified

[2] Based on current examples where the system is in use ten hours a day, there are two color changes at 1/2 hour each, and there is 50% system degradation for an hour before each change.

command functions and operation of the contingency management functions. A small portion of the flight-following data base became classified and this portion had to be processed manually to avoid contaminating the entire data base. Addition of a processor at MAC has eliminated the requirement for manual processing of classified information, but manual re-entry and integration of information are still necessary. Consequently, even though additional equipment is available, MAC lacks an integrated and responsive svstem for managing its aircraft force.

A second difficulty is the integration of intelligence and operations data. Such integration is required for responsive battle management, but it must be done so as not to jeopardize intelligence sources. It is often impossible to clear all system users for the intelligence data, so manual intervention is used: a cleared intelligence officer hands a subset of the data to the operations element. However, as automated, timely integration of such data becomes necessary, this option becomes unacceptable, and a direct technological solution to the multilevel security problem must be found.

Requirements Summary

What has been said summarizes the major effects of the current practices that attempt to meet the requirement for computer security. Experience has indicated that the cost may run ten to twenty percent of the total operating cost of the Air Force computer installations that process classified data -- from $20 to $40 million per year. Operationally, some requirements are met by buying additional equipment and facilities, but a significant requirement for real-time information sharing is arising and this requirement cannot be met even by buying such equipment.

THE TECHNICAL PROBLEM OF MULTILEVEL SECURITY

The case against relying on the costly, restrictive procedures outlined above is strong. Economic and operational considerations argue for developing the ability to process an arbitrary mix of classified and unclassified information simultaneously with a single computer, serving cleared and uncleared users and relying on the computer´s and operating system´s internal controls to enforce security and need-to-know requirements. Such a computer would be operating in a multilevel security mode; the presence of uncleared users (or users at unsecured terminals) would define an open multilevel mode.

Unfortunately, however, the costly procedures used today continue to be necessary -- made by the inability of current hardware-software systems to protect the information they process. The only sound assumption that can be made about a current computer system concerning information protection is that any program that runs on the system can access any information physically accessible to the processor, and can retrieve, alter or destroy the information as the programmer wishes.

While the assumption stated above may appear radical, it is amply supported by facts and experience. On numerous occasions, programmers have conducted formal or informal projects aimed at testing the security of operating systems by penetration -- by writing programs that obtain access to information without authorization. ESD personnel have directly participated in several of these penetration projects and have observed the results of others. In each case the result has been total success for the penetrators. The programmers involved in these efforts have not been "insiders" but simply competent system programmers armed with user and (sometimes) system-level documentation for the computer and operating system under test.

No real hostile penetrations of military computers processing classified information have been reported. However, this absence is due to the protective procedures of the external sort just described, not because it is difficult to make a programmed penetration against these computers.

Given experience in the penetration of computer systems, one might ask "Why not simply modify the operating system programs to correct the flaws that permit the penetration?" Two problems prevent this approach (often referred to as "patching holes") from being effective. The first is that in many cases operating system or application programs will not work if a hole is patched. Thus, correcting a security flaw may render the computer system inoperative unless a long, costly series of program modifications is made. This problem is compounded at the practical level by the fact that complex, expensive program modifications, intended to patch existing operating system holes, may themselves introduce new holes in previously sound areas.

The second problem, a fundamental one in the field of multilevel computer security, is that of completeness. Even if every hole that allowed a known penetration approach to work were repaired, one still could not consider the resulting operating system secure, because a given collection of penetration programs exposes only the holes that those programs exploit. Short of constructing the astronomically large set of all possible penetration programs, one can make no statement at all about undiscovered holes, or about the penetration programs that would exploit them.

The problem of completeness, as stated above, may prompt the reader to object that completeness is not necessary. Nowhere else is perfect security required; physical, personnel and even communications security measures have finite probabilities of penetration. Is it not then possible to accept a degree of computer security less than a hundred percent? Unfortunately, the usual analogy between operating system security problems and those of physical, personnel or communications systems does not hold. If even one error in an operating system program allows a penetration program to work, that program will work every time it is executed -- typically retrieving without detection any information accessible to the computer. The probability of a successful penetration is then unity; the level of security, zero. The likelihood that a hostile agent will write the penetration program is the only uncertainty. This likelihood is hard to assess, since it depends on the agent's motivation and competence. However, experience with penetration tests leads to the conclusion that the penetrator's chances of success are very high.

Restricting access to operating system documentation is not a safeguard. Although concealing the structure of the operating system may seem to obscure the weaknesses of the security controls, such a primitive encoding scheme does not effectively deter penetration; knowledge of the basic processor hardware and of any standard operating system is an adequate starting point for the penetrator's efforts.

A final point about the vulnerability of current computer systems concerns the cost of penetration. Most penetration efforts have been completed successfully with very few (perhaps two) man-months of effort. Typically, the bulk of the effort expended is directed toward exploitation -- finding information to be retrieved and building programs to retrieve it. Development of the basic approaches that assure successful penetration has usually required only a man-week or two. In comparison, the effort expended in patching operating system holes is rumored[3] to be in the tens or hundreds of man-months.

This brief overview of the technical problem of multilevel computer security is not intended to portray the problem as hopeless. Rather, the intention is to show that the problem is difficult and that the alternative of patching holes in current operating systems is futile. The next section introduces a unified technical approach to development of secure computer systems.

[3] Most agencies that have performed such patches are reluctant to report costs.

SECTION III

A UNIFIED TECHNICAL APPROACH TO MULTILEVEL COMPUTER SECURITY

This section introduces the foundation of ESD computer security development effort. Its three subsections describe the history and origin of the technical approach; briefly summarize the approach and its main implications; and discuss the technique for verifying the security of a computer system that solves the completeness problem.

THE COMPUTER SECURITY TECHNOLOGY PANEL

In 1970, the Air Force Data Services Center (AFDSC) asked the Electronic Systems Division to support development of open multilevel secure operation for AFDSC's Honeywell 635 computer systems. ESD and MITRE personnel shortly reached conclusions substantially identical to those given above: that no set of modifications to the 635's operating system would render it suitable for multilevel operation, much less for open operation with uncleared users and terminals.

To determine the reasons for this difficulty, and to identify ways of solving future multilevel security problems, the Air Staff directed ESD in 1972 to convene a computer security technology planning study panel. The panel was composed of recognized experts from industry, universities, and government organizations and operated under a contract from ESD to James P. Anderson Company. It was tasked with reviewing projected Air Force needs, identifying, recommending a technical approach, and preparing a development plan for a coherent approach to attacking the problems of multilevel computer security. The panel was supported by a requirements working group of computer system staff officers from ten Air Force commands.

The panel's report [2] identified the problem of completeness and recognized the futility of "patching holes" in existing operating systems. It recommended a technical approach that starts with a model of an ideal secure system and refines it through various levels of design into hardware-software mechanisms that implement the model. The report also described an earlier version of the development effort described herein.

THE REFERENCE MONITOR

The basic component of the technical approach proposed by the security technology panel is the reference monitor -- an abstract mechanism that controls access by subjects (active system elements) to objects (units of information) within the computer system. Figure 1 schematically diagrams the relationships among the subjects, objects, reference monitor, and reference monitor authorization data base, and gives examples of typical elements. An implementation of the reference monitor abstraction permits or prevents access by subjects to objects, making its decisions on the basis of information contained in the reference monitor data base. The implementation both mechanizes the access rules of the military security system, and assures that they are enforced within the computer.

The security technology panel stated that, to be the basis for a multilevel secure computer system, a mechanism that implements a reference monitor must meet three requirements:

> Completeness -- the mechanism must be invoked on every access by a subject to an object.
>
> Isolation -- the mechanism and its data base must be protected from unauthorized alteration.
>
> Verifiability -- the mechanism must be small, simple and understandable so that it can be completely tested and verified to perform its functions properly.

These requirements, and the need for efficiency, demand that the reference monitor implementation include hardware as well as software, because software validation of every access by a subject to an object would add intolerable complexity and overhead to the reference monitor. Particular hardware features considered essential to implementation of a secure system include segmented memories, processors with multiple domains of execution, and positive control of all I/O devices.

The hardware-software mechanism that implements the reference monitor abstraction is called the security kernel. When the computer hardware is predetermined, the software that must be designed to implement the reference monitor abstraction is frequently referred to as the security kernel for that computer.

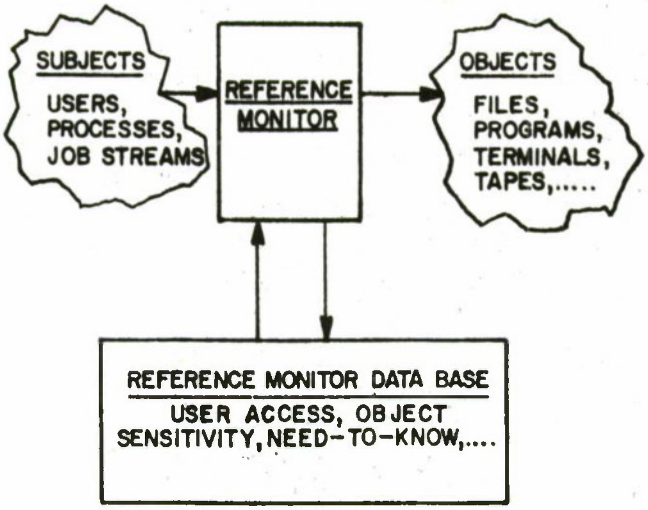

Figure 1. Reference Monitor

MODELS AND TECHNICAL VALIDATION

Recognizing the importance of the model of an ideal system recommended by the security panel as a starting point, ESD initiated development of a mathematical model of computer security in 1972. Preliminary efforts were performed by ESD [3] and subsequent contributions were made by The MITRE Corporation and by Case Western Reserve University. The model specifies requirements for the operation of a security kernel. The security requirements for the the model are taken directly from the Defense Department regulations on handling sensitive information (DoD Directive 5200.1-R).

The completed model of secure computer systems [4] [5] [6] represents a secure computer system as a finite-state mechanism that makes explicit transitions from one secure state to the next. The state of the system is defined by:

> the classifications and compartments of all subjects and objects;
>
> the need-to-know relationships of subjects and objects; and
>
> subjects' current ability to access objects.

The rules of the model formally define the conditions under which a transition from state to state can occur. The rules are proven to allow only transitions that preserve the security of information in the system.

A significant property of the model is that all but a special collection of proven and trusted programs are restricted from writing information at a lower classification (or proper subset of compartments) than they read. The restriction prevents information obtained at the higher level from being transferred to a lower level where it can be accessed illegally. This property, referred to as the *-property or confinement property, eliminates the need to verify that all programs (such as editors and utility routines) do not act as "Trojan Horses"[4] and downgrade classified information.

For some time after the basic security model was developed, there was doubt as to the appropriate technical approach to providing

[4] A Trojan Horse is a computer program that is typically developed by one individual for use by another. When the program is operating on behalf of the intended user, it accesses that user's sensitive data, and makes it available to the program's author. [7]

complete assurance that the security kernel behaves as the model requires. In 1973 it was recognized that the work of Price [8] identifies a methodology for providing the required assurance. This methodology involves preparing a formal specification for each function of the security kernel. The collection of specifications is then proven to be internally consistent and to implement the rules of the model. The descriptions of the functions in the specification language are close to a programming language and facilitate proof or verification of the code that implements the specified kernel design.[5]

While the basic methodology developed by Price applies to validation of small security kernels (up to perhaps 1000-line computer programs), the consistency proof may become cumbersome for larger kernels. In addition, the Price methodology is inadequate for dealing with some aspects of implementation; the presence of system wide variables, for instance, precludes a proof of security. Therefore, a levels of abstraction approach that is based on a structured specification and proof technique and that divides the specification modules into manageable subsets is being employed in addition to the basic Price methodology. [12]

The paragraphs above have summarized the basic elements of ESD's approach to the design and technical validation of secure computer systems and security kernels. While the administrative certification that a computer is secure must be based on formal policy, only a technical validation approach, such as outlined, and shown schematically in Figure 2, can be an adequate basis for such certification policy.

[5] A more detailed description of the validation methodology has been prepared by MITRE and is contained in [9], [10] and [11].

IA-49,399

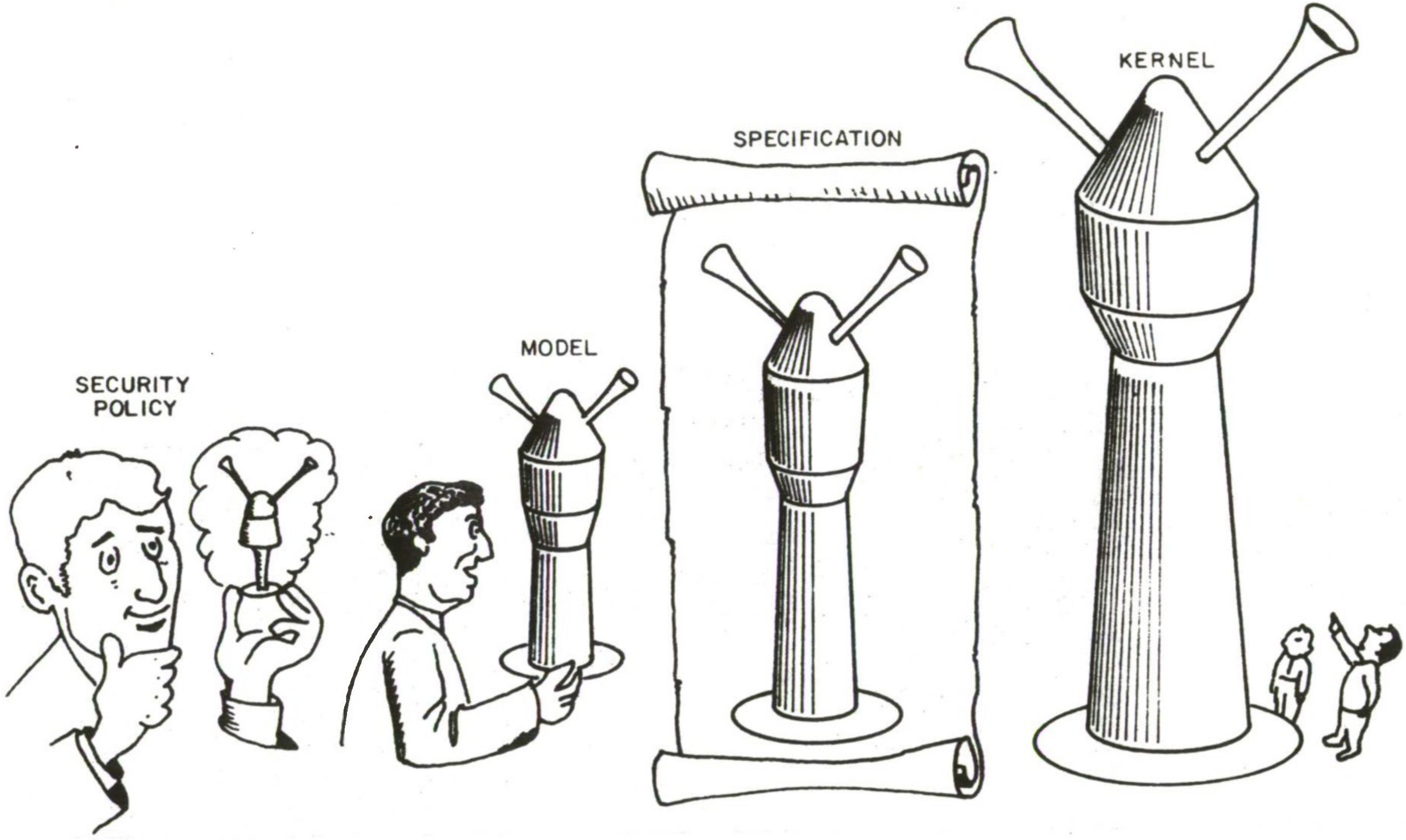

Figure 2 INCREMENTAL VERIFICATION

SECTION IV

SECURE COMPUTER SYSTEMS

This section presents an overview of two major secure computer system developments that apply the technical approach described above. They are aimed at providing the Air Force with immediate improvements in its ability to meet computer security requirements, and with long-term solutions to very general computer security requirements. The first is the "brassboard security kernel," a general-purpose security kernel for an off-the-shelf minicomputer. The second is a security kernel for Multics, a large general-purpose computer system. Figure 3 shows conceptually the basis for security in computer systems.

THE BRASSBOARD SECURITY KERNEL

In order to demonstrate the viability of the security kernel technology, ESD directed The MITRE Corporation in January 1973 to begin implementing a prototype security kernel for the Digital Equipment Corporation PDP-11/45, a relatively large [13], moderately priced minicomputer. This kernel was initially intended to serve as the base for a front-end communications processor for use with a secure general-purpose computer system to be developed later. However, it was soon realized that the kernel could also support stand-alone secure computer applications requiring only a minicomputer and, most important, could serve as a "brassboard" to prove out the model and kernel concepts long before developing a kernel for a large general-purpose system. While a kernel for a large, general-purpose computer need not be much larger than that for a minicomputer, the amount of non-security operating system software needed to effectively use the large system is far greater. Consequently, this phased approach was considered desirable.

The kernel design for the PDP-11/45 was developed by applying Dijkstra's principle of levels of abstraction [14] to separate the parts of the kernel that implement the security rules, objects and subjects required by the model. The kernel implements separate sequential processes that can cooperate and communicate in accordance with the rules of the model. This kernel design creates a very basic secure environment upon which operating systems and application programs can be implemented. The access of users (subjects) to information (objects) in such a kernel-based system would conform to the specified security rules since all system and applications software is running on top of the security kernel.

IA-49,400

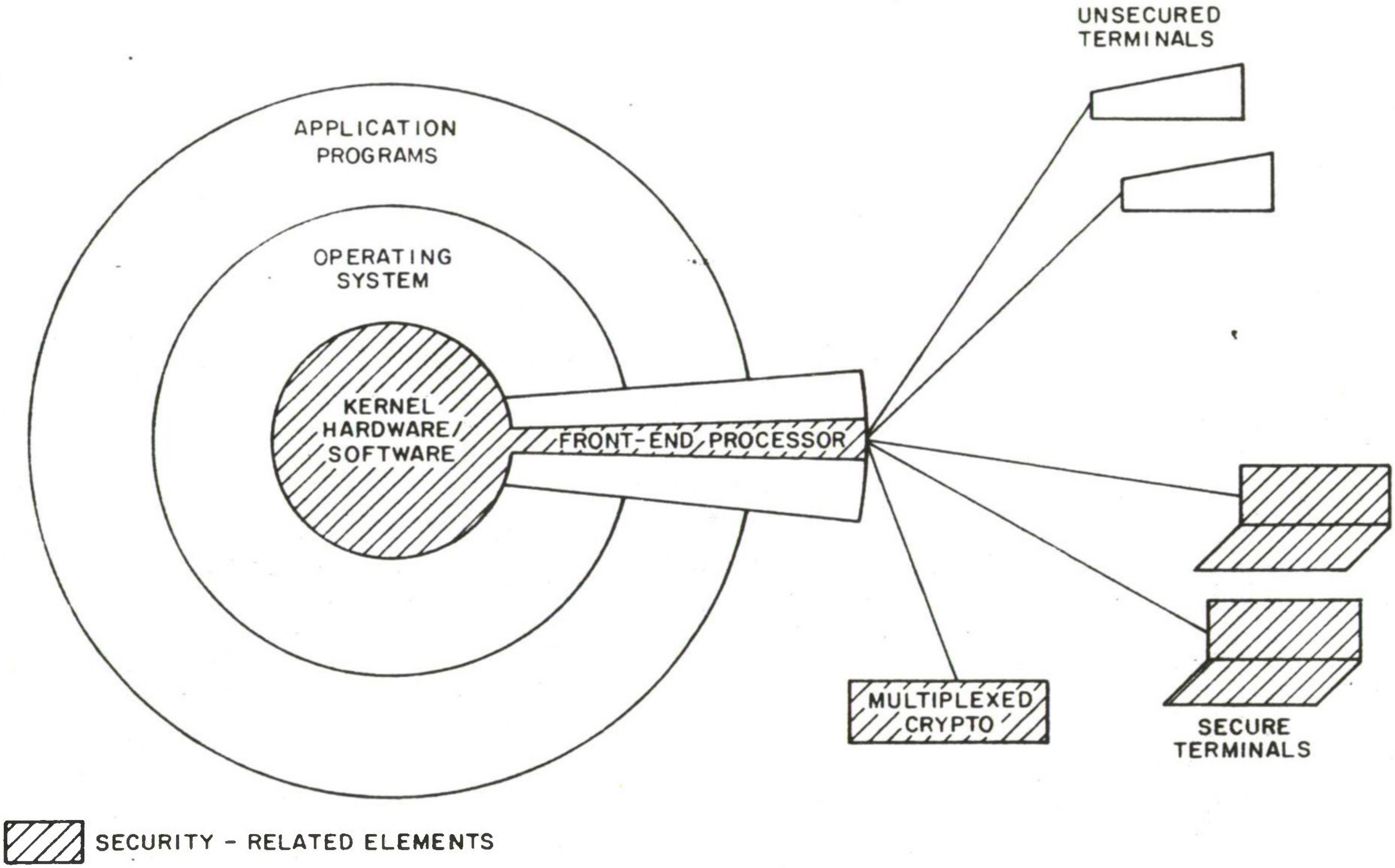

Figure 3 SECURE COMPUTER SYSTEM

Design of the PDP-11/45 security kernel was completed in early 1974. The kernel programs were implemented in a higher-order language (the Project SUE Systems Language) and compiled and tested in spring 1974. Verification of the brassboard kernel's security began in 1974 with completion of formal specifications. [15] Proof that the specifications are consistent and implement the model was initiated then, and a single module was proven to verify the feasibility of the proof method. A complete verification that the top level of the formal specification did in fact constitute a valid interpretation of the mathematical model was finished in late 1975 and, concurrently, a more detailed lower level specification was proved to correctly implement the upper level. [11]

Efforts to exploit this brassboard system and thereby demonstrate the potential of security kernel technology have centered in two areas: adaptation of the UNIX operating system to the 11/45 kernel, and constructed of a multilevel file management system to simulate the automated handling of intelligence data. The UNIX effort was initiated in 1975 and should be completed by late 1976. Under this project, there will be a reimplementation of the kernel[6] and implementation of a UNIX emulator to run on the kernel. The objective is to produce a secure, efficient system that produces a user interface consistent with that of an existing system. In particular, a user interface is being sought that is rich enough to facilitate use of the system in a production or development environment. The design is being strongly influenced by anticipated applications.

As a practical demonstration of security kernel technology, a MITRE project built a secure, multilevel, file management system on the PDP-11/45 kernel. Two scenarios using this file system have been developed. The first of these scenarios uses a text editing capability to show how a multilevel data base can provide for data storage, manipulation, and retrieval, in a multilevel user environment, while protecting all classified information from unauthorized access. The second demonstration employs an air surveillance data correlation scenario that permits precisely controlled, selective downgrading of classified track data based on the informed judgment of a downgrading officer. The system being demonstrated allows users to access the widest possible range of information on the system (restricted only by their maximum clearance) yet prevents the unauthorized user from accessing any classified information not specifically downgraded.

[6] Rewriting is primarily necessary for reasons of efficiency. In addition, use of a more widely known implementation language is planned in order to broaden the circulation of the final product.

In summary, the PDP-11/45 security kernel provides an early demonstration of the feasibility of building a security kernel that implements the model. Each step in the sequence from model to kernel code is subject to proof or verification. Both the brassboard kernel and the reimplemented kernel with UNIX interface will be available for performance tests, penetration tests (which will undoubtedly be desired even though their failure is not a proof of security), inspection, review and application.

THE MULTICS SECURITY KERNEL

While the security kernel for the PDP-11/45 constitutes a small secure system, Air Force commands and users such as MAC, SAC, and TAC and AFDSC need large multilevel secure computers. The reference monitor concept must be demonstrated to be feasible in an efficient, as well as secure large resource-sharing system. This demonstration is necessary to show that systems based on the reference monitor concept can provide a viable solution to meeting all Air Force ADP requirements (not just those for security). For these reasons, ESD has set as a goal development of a security kernel and operating system for a large computer.

The Honeywell 6180 (or successor 68/80) computer and its Multics operating system were chosen as the base for a secure large-scale prototype, for two principal reasons. First, the 6180 hardware supports a segmented virtual memory and multiple protection domains in a way that makes it well-suited to support a kernel. In fact, a study of hardware architectures for security completed in mid-1974 [16] determined that the 6180 was the off-the-shelf large computer best suited to support a security kernel.

The second reason for choosing the 6180 and Multics relates to the Multics operating system. Multics is written to implement a segmented virtual memory, and to use that segmented memory where possible within the operating system. Thus the existing user programs and many operating system programs are compatible with the environment that a security kernel is expected to provide. This fact should significantly reduce the cost of a Multics-based secure system, for it appears that the (non-security related) operating system software, rather than the security kernel, will be the major cost component in any kernel-based secure computer system.

Initial steps toward developing a secure system based on Multics were taken in conjunction with development of a Multics operating system for use in a two-level (Secret and Top Secret) environment at the Air Force Data Services Center. This system´s design is aimed at providing security controls based on the military access rules, but it does not attempt to eliminate completely the prospect of hostile penetration. The risk of penetration is to be reduced primarily by procedures and by personnel and environmental controls, rather than by the Multics hardware and software. The implementation of the access rules in the Data Services Center Multics was based on the secure system model described in the previous section, but no attempt was made to define a security kernel for the system.

The design of the Air Force Data Services Center Multics was begun in late 1973 and completed in mid-1974. [17] Implementation was completed in 1975 and the system is currently in operation. Information on this system's utility and security has proved useful to the design of a Multics security kernel. Furthermore, the user interface of the Data Services Center Multics has been designed to resemble that of a kernel-based system, so that the transition from the Data Services Center Multics to a kernel-based Multics will be relatively easy, and so that operational experience will be available to guide the Multics kernel design. The Data Services Center Multics had no significant (<1%) overhead although it has all the security checks of a kernel-based system. [18]

Design of a Multics kernel began in September 1974 with a concentrated one-month session involving staff members from ESD, the MITRE Corporation, Honeywell and the Massachusetts Institute of Technology (a co-developer of Multics). The resulting kernel design includes a segmented and paged virtual memory similar to that of the standard Multics operating system, with security controls and organization similar to those in the PDP-11/45 "brassboard" kernel. The input/output system required by the kernel is based, in part, on the use of a minicomputer front-end processor with its own kernel to provide a secure flexible interface to external devices. [19] The 11/45 brassboard was initially considered as a candidate for this front-end processor but was found to be inadequate.[7]

Design and implementation of a prototype secure Multics (including a secure front-end processor) is a major goal of the computer security development program. Honeywell has been involved in the prototype effort since July 1974, via a cost-sharing contract with ESD. MITRE is acting as the system engineer for this effort. To date, Honeywell has prepared a specification for the secure front-end (SFEP) hardware and is planning to make a hardware prototype available by the end of 1976. Design of the SFEP kernel and software will be finished in 1976 and a prototype SFEP completed in 1978. With MIT as a subcontractor, Honeywell is preparing a complete set of formal

[7] Particular problems with use of the 11/45 as a front-end are: the small number of segments per protection domain, the slowness of process swapping, and the inconveniences in implementing security controls for I/O devices.

specifications for the Multics kernel guided by MITRE-supplied preliminary specifications. In addition, revisions to the (non-security) Multics operating system that will provide a complete, usable environment are being defined. Another subcontractor, SRI, will assist in developing the validation techniques and provide a proof of the correspondence between the levels of the formal specifications. A top-level specification of the Multics kernel will be completed in mid 1977 and a preliminary demonstration of the secure prototype (with SFEP) is scheduled for 1978. A final version of the prototype will be available in early 1980.

SECTION V

SUMMARY

This document has described the problem of multilevel computer security and a technological basis for its solution. Section II reviewed the current alternatives for processing classified information with ADP systems, and outlined the major economic and operational impacts of those alternatives.

The reference monitor concept introduced in Section III offers a technological basis for security controls whose effectiveness can be verified. The secure systems described in Section IV apply the reference monitor concept to meet the requirements of Air Force users. The PDP-11/45 security kernel is the heart of a small secure system that can be used in the near term. That kernel is based on a mathematical model and is already in experimental use. Its security is now being verified by a rigorous formal process. The Multics security kernel will provide the prototype of a large multilevel system for use in command-control, administrative and intelligence applications. These tasks and the others that constitute the ESD Computer Security Development Program are summarized in Appendix I.

The reference monitor concept has been brought from an academic abstraction to a basis for security in real systems. The development tasks exploit the concept in an orderly manner -- first by developing prototypes of secure systems that apply the concept, and then by transferring the techniques proven by the prototypes to operational systems in the field. The basic approach has been shown to be technically sound, and its development will allow the Air Force to meet its pressing requirements for secure multilevel computing.

APPENDIX I

OUTLINE OF THE DEVELOPMENT EFFORT

The tasks of the development effort will produce techniques, prototypes and application aids aimed at equipping Air Force computer users with the capability to do efficient secure multilevel computing. They should result in an immediate improvement in Air Force users ability to meet their computer security requirements as indicated in Figure 4. The intent of this appendix is to present an overview of each of the more than sixty component tasks that make up the effort, and to indicate how they fit together.

For the purpose of this appendix, the tasks have been divided into five groups:

Prerequisites

Secure general-purpose system development

Technology transfer

Application aids development

Secure computing environment development group

Figure 5 provides an overview of the prototype development efforts and Figure 6 depicts the relationship of the groups and tasks. Reference to this figure may prove helpful when reading their descriptions.

PREREQUISITES

The prerequisites group includes initial tasks necessary to achieve multilevel secure computing capabilities. The tasks develop the plans, theories, technology and demonstrations necessary to solve the multilevel computer security problem. Most of the tasks have already been completed and are discussed in the main body of this document.

Specific tasks in the prerequisite group include:

Task 1 -- Panel of Experts: formation and operation of the ESD computer security technology panel. (Completed)

IA-49,401

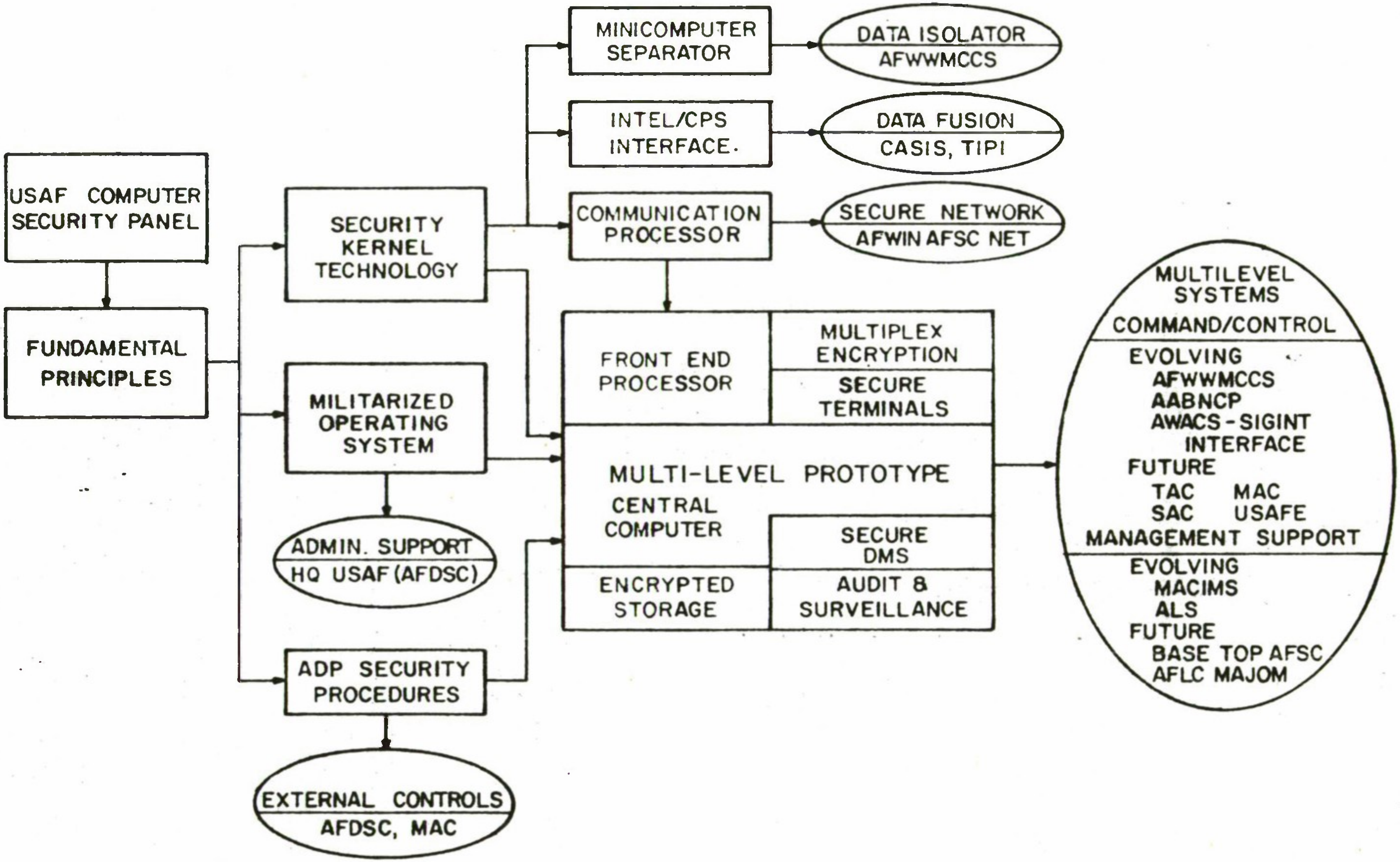

Figure 4 PROGRAM STRUCTURE

IA-49,402

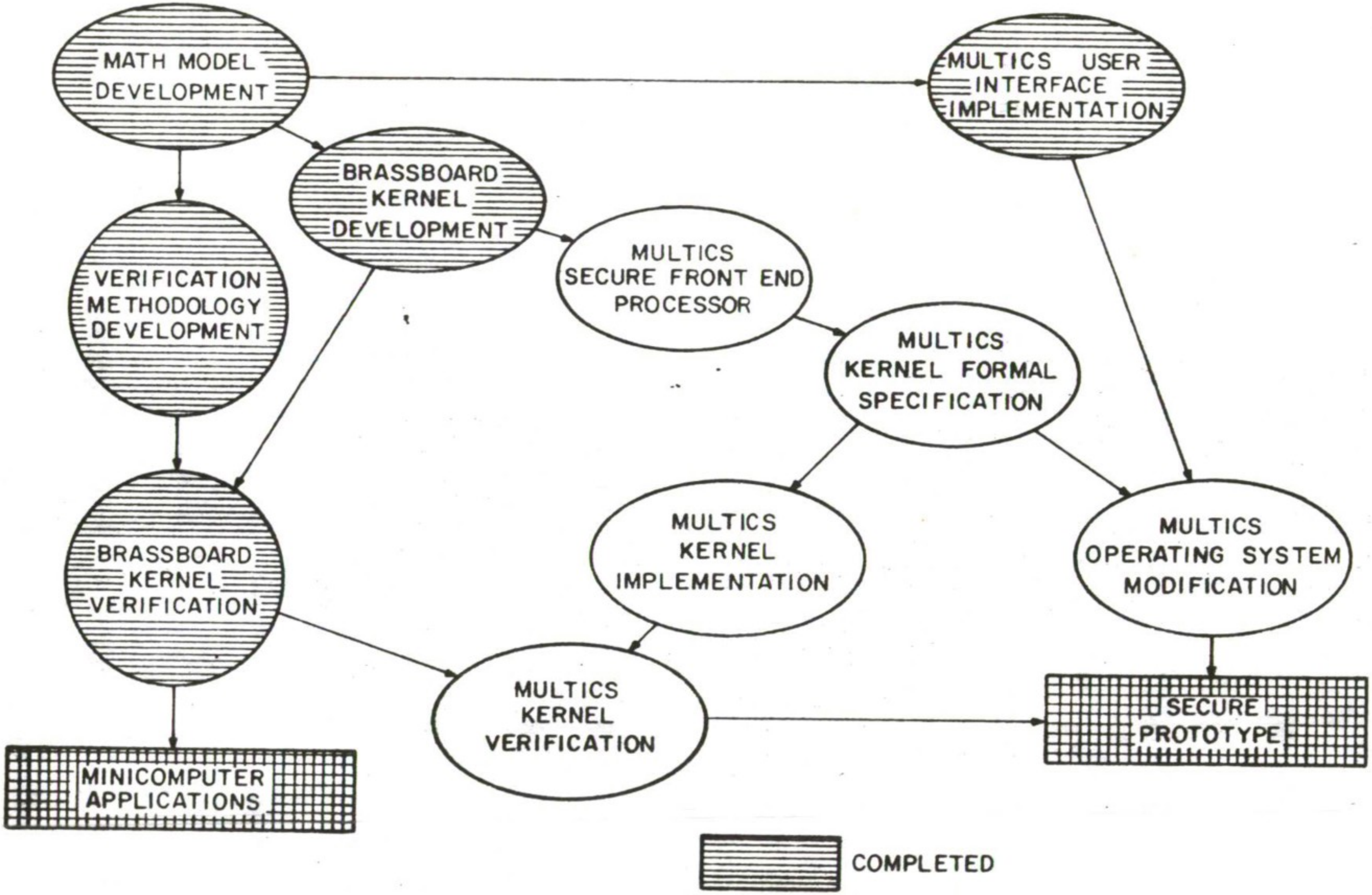
MATH MODEL DEVELOPMENT
VERIFICATION METHODOLOGY DEVELOPMENT
BRASSBOARD KERNEL DEVELOPMENT
BRASSBOARD KERNEL VERIFICATION
MINICOMPUTER APPLICATIONS
MULTICS SECURE FRONT END PROCESSOR
MULTICS KERNEL FORMAL SPECIFICATION
MULTICS KERNEL IMPLEMENTATION
MULTICS KERNEL VERIFICATION
MULTICS USER INTERFACE IMPLEMENTATION
MULTICS OPERATING SYSTEM MODIFICATION
SECURE PROTOTYPE
COMPLETED

Figure 5 PROTOTYPE DEVELOPMENT OVERVIEW

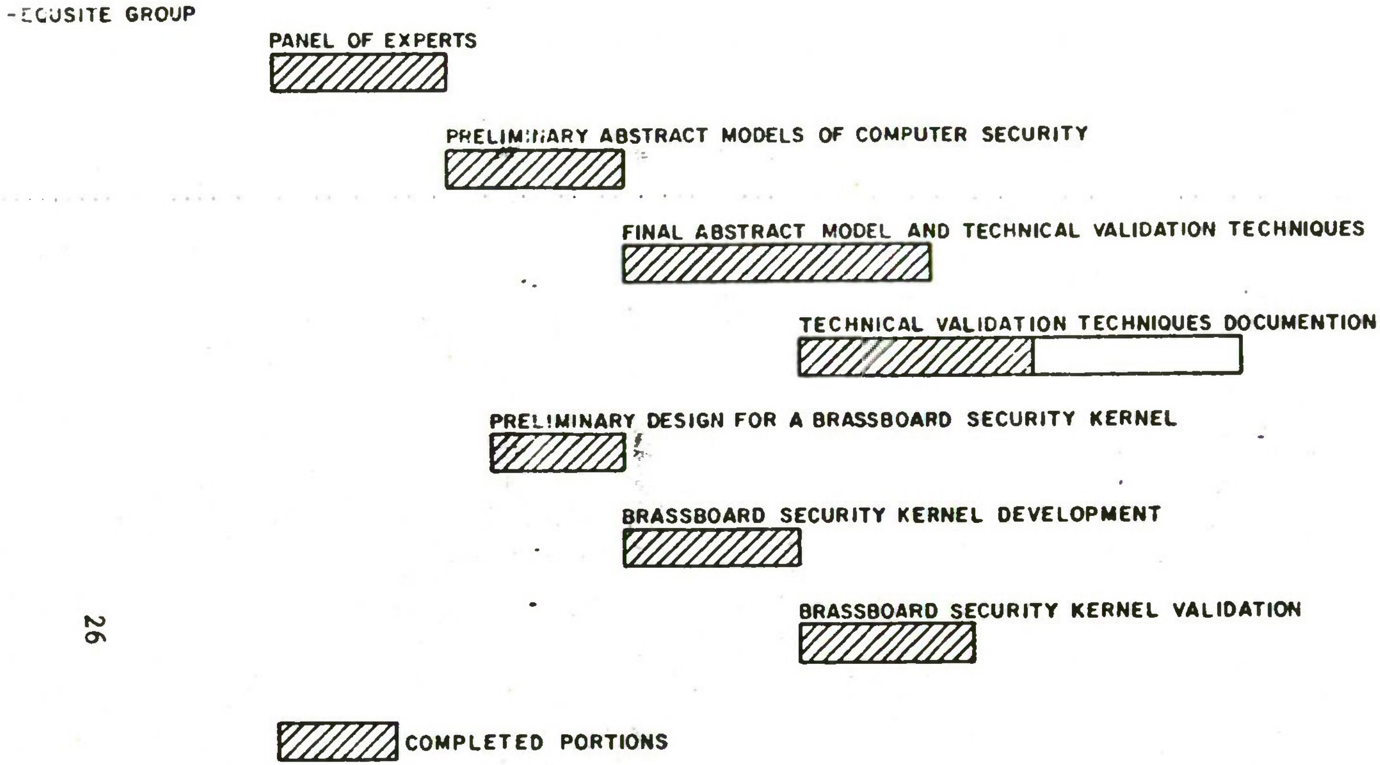

Figure 6a ADP SECURITY DEVELOPMENT PROGRAM SCHEDULE

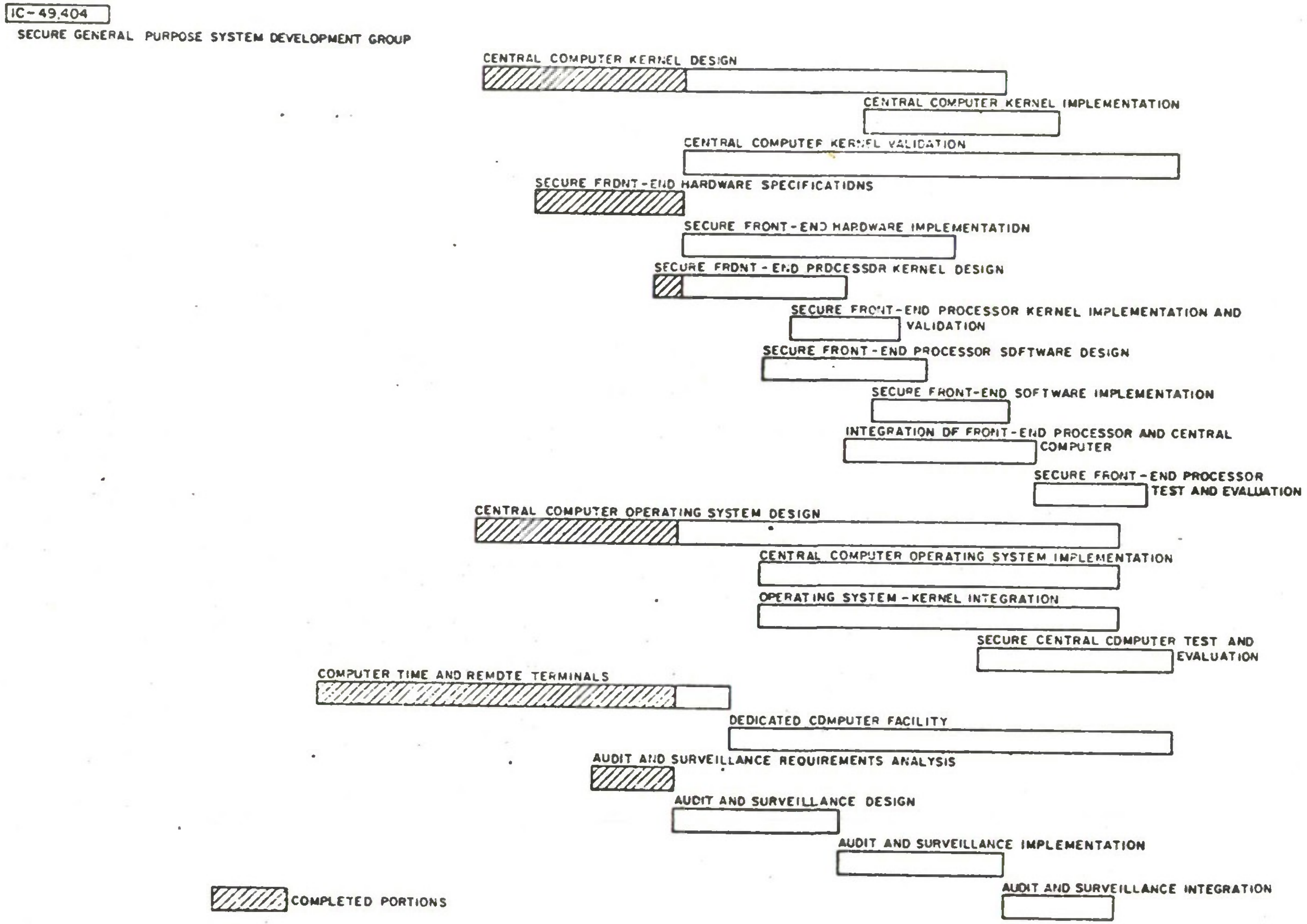

Figure 6b ADP SECURITY DEVELOPMENT PROGRAM SCHEDULE

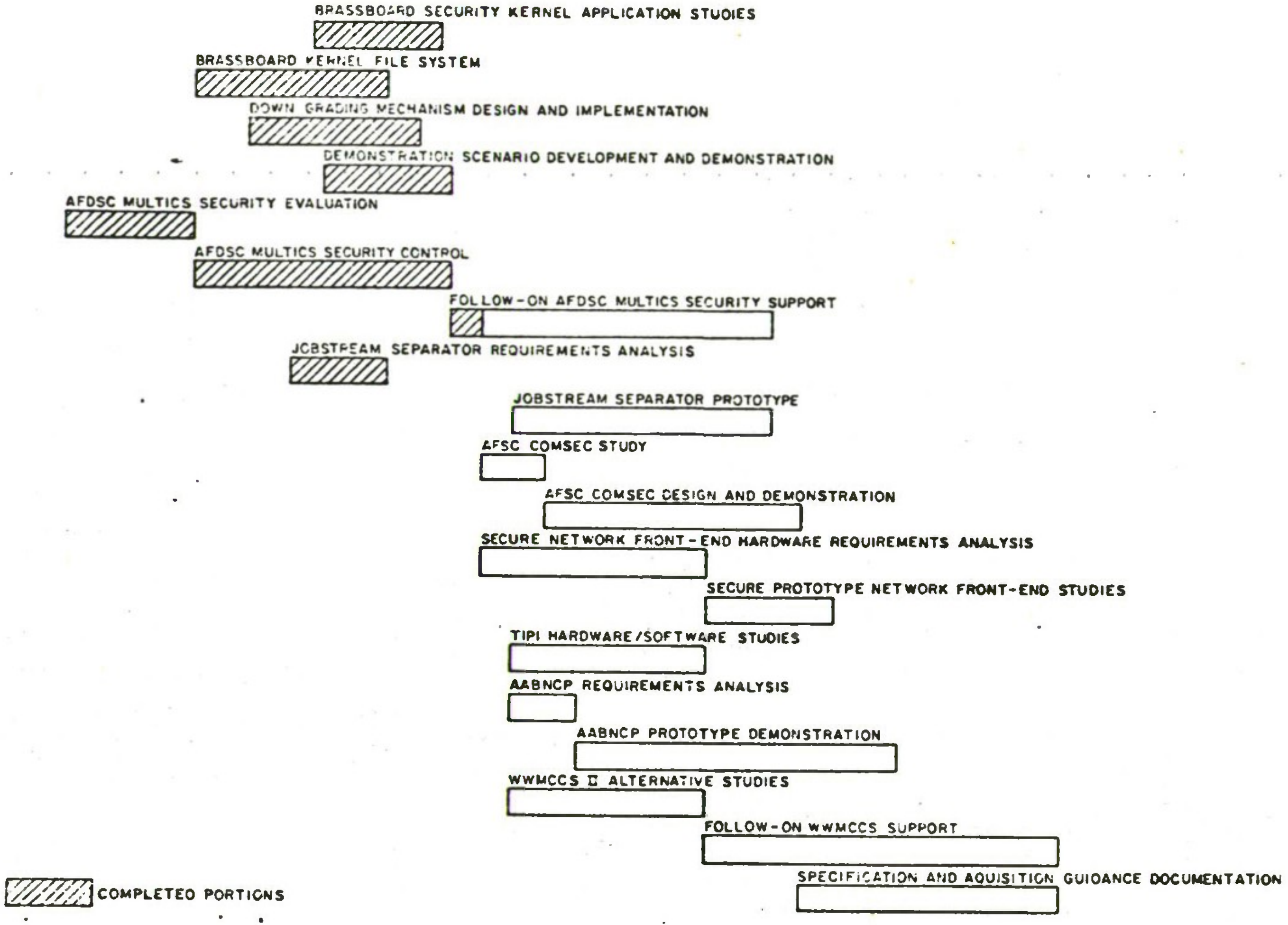

Figure 6c ADP SECURITY DEVELOPMENT PROGRAM SCHEDULE

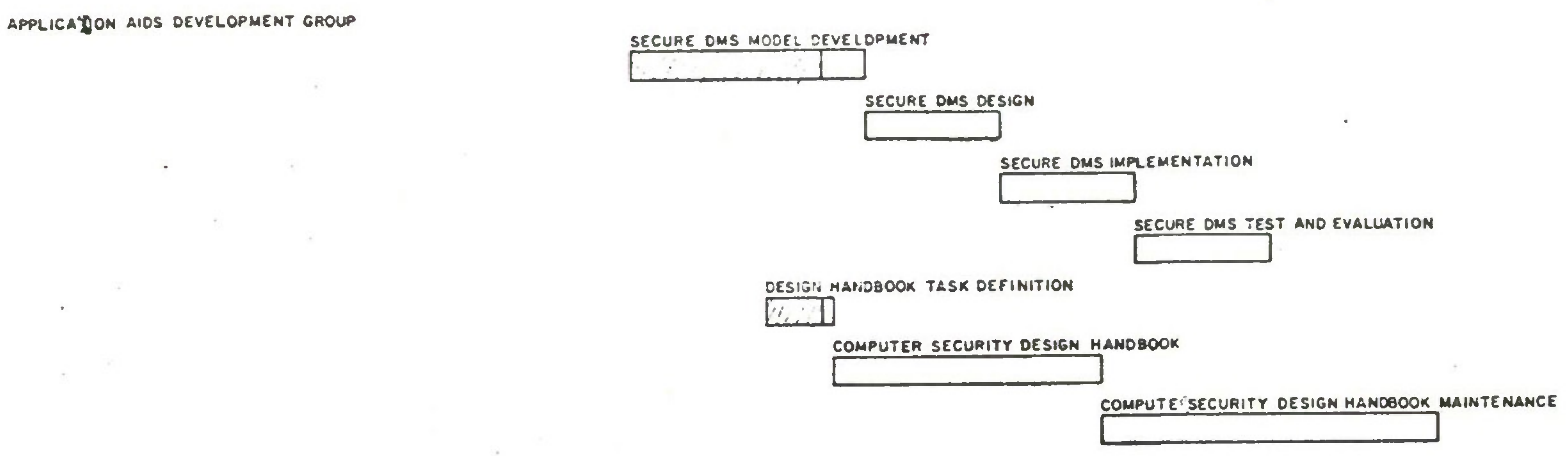

Figure 6d ADP SECURITY DEVELOPMENT PROGRAM SCHEDULE

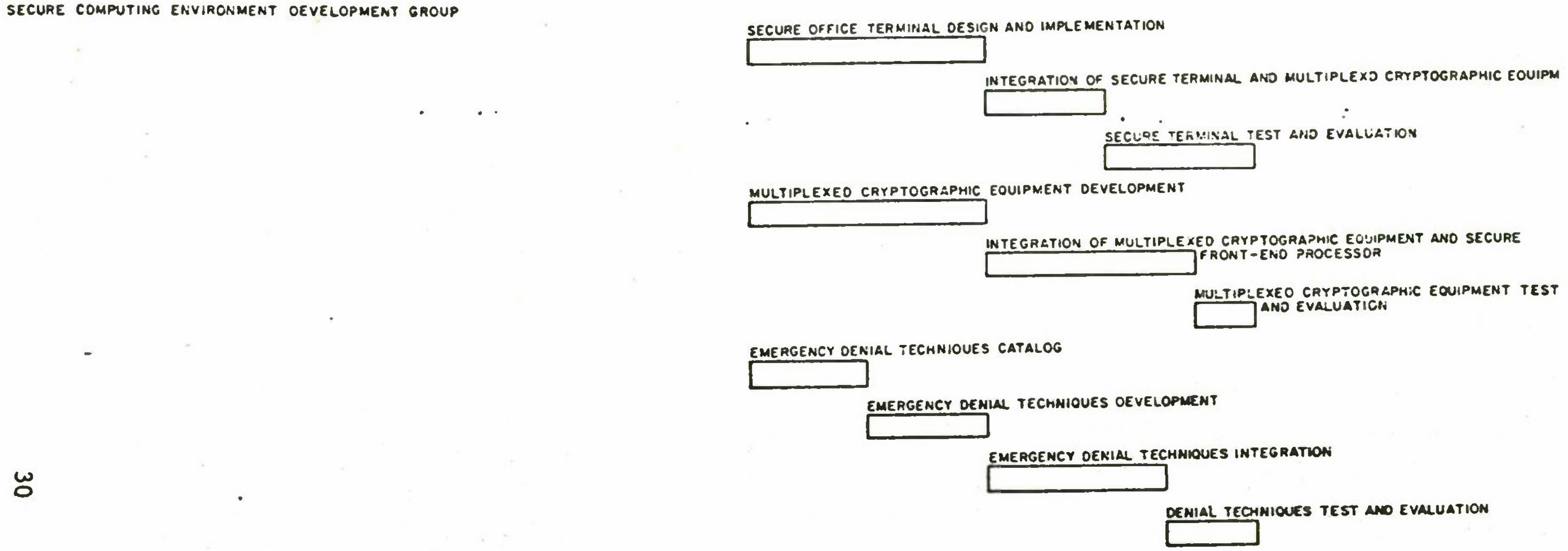

Figure 6e ADP SECURITY DEVELOPMENT PROGRAM SCHEDULE

Task 2 -- Preliminary Abstract Models of Computer Security: The preliminary model task involved the early phases of the security model developments by ESD, MITRE and Case Western Reserve University. (Completed)

Task 3 -- Final Abstract Model and Technical Validation Techniques: The final models describe objects that correspond to segments in a storage hierarchy. This task also addresses development and application of technical validation techniques that can be applied to kernel module formal specifications. (Completed)

Task 4 -- Technical Validation Techniques Documentation: provides formal documentation and tools for verifying that a security kernel corresponds to the security model. (In progress)

Task 5 -- Preliminary Design for a Brassboard Security Kernel: developed the first design iteration for the PDP-11/45 security kernel. (Completed)

Task 6 -- Brassboard Security Kernel Development: completed the design and implementation of the security kernel for the PDP-11/45. (Completed)

Task 7 -- Brassboard Security Kernel Validation: proceeds with the proofs and verifications required to effect technical validation of the Brassboard Security Kernel. (Completed)

SECURE GENERAL-PURPOSE SYSTEM DEVELOPMENT

The secure general-purpose system development group takes the models, tools and concepts prepared by the prerequisite group and reduces them to practice by developing a large-scale, general-purpose secure system. The product is a prototype of a secure large-scale computer system (based on the existing Multics system) suitable for field use and capable of serving as a guide for Air Force users who have a requirement for such systems. The tasks in this group cover development and technical validation of kernels for the secure computer system and its front-end processor, and modification of the operating system software to provide a useful computing environment outside the kernels, and auditing of user actions in a secure environment to enforce requirements for user accountability and responsibility.

Task 8 -- Central Computer Kernel Design: The mathematical model and brassboard kernel design are the foundation for design of a kernel for a secure general-purpose central computer. This task develops the design of a formal specification for a kernel for the Honeywell 6180 processor. (In progress)

Task 9 -- Central Computer Kernel Implementation: Given a design for a central computer security kernel, this task develops the code that implements the security kernel. (Planned)

Task 10 -- Central Computer Kernel Validation: proceeds with the proofs and verifications (also penetration tests, if desired) required to effect validation of the kernel for the central processor of the secure general-purpose system. (Proceeding in conjunction the kernel design of Task 8.) (Planned)

Task 11 -- Secure Front-End Hardware Specification: specifies a hardware architecture that provides a basis for implementation of a secure front-end processor for the secure central computer. This architecture must be capable of supporting its own security kernel. (Completed)

Task 12 -- Secure Front-End Hardware Implementation: provides the hardware for the secure front-end processor. (In progress)

Task 13 -- Secure Front-End Processor Kernel Design: will produce in the design of a certifiable security kernel for the front-end processor. Formal specifications will be used to define and aid in the kernel verification. (In progress)

Task 14 -- Secure Front-End Processor Kernel Implementation and Validation: The kernel designed in Task 13 is implemented on the hardware made available by Task 12, and the proofs and verifications necessary to effect validation are performed. (Planned)

Task 15 -- Secure Front-End Processor Software Design: will produce a design for all other software necessary to interface the front-end processor with the central computer. (Planned)

Task 16 -- Secure Front-End Processor Software Implementation: implements all non-kernel software in accordance with the design developed under Task 15. (Planned)

Task 17 -- Integration of Front-End Processor and Central Computer: integrates the front-end processor and the central computer into a cooperating unit. Special attention is paid to the interaction of the two processors' security kernels. (Planned)

Task 18 -- Secure Front-End Processor Test and Evaluation: performs the functional test and evaluation of the secure front-end processor in an environment that includes a secure central computer and secure communications peripherals. (Planned)

Task 19 -- Central Computer Operating System Design: The operating system for the secure central computer must exploit the environment provided by the kernel. This task designs a suitable operating system based as much as possible on the existing Multics operating system. (In progress)

Task 20 -- Central Computer Operating System Implementation: modifies the Multics operating system to work with the kernel, based on the design prepared by Task 19. (Planned)

Task 21 -- Operating System-Kernel Integration: integrates the central computer security kernel and operating system. (Planned)

Task 22 -- Secure Central Computer Test and Evaluation: tests and evaluates the utility, efficiency, and acceptability of the secure general-purpose computer in a user environment. (Planned)

Task 23 -- Computer Time and Remote Terminals: represents the requirement of the secure general-purpose system development group for timesharing access to a Multics computer system. Such access is required during the early phases of the central computer kernel and operating system design and development. (In progress)

Task 24 -- Dedicated Computer Facility: Once implementation of the central computer kernel and operating system begins in earnest, a dedicated secure facility is required to support development, testing and kernel storage. While such a facility can support users other than those involved in the secure system development, the nature of the kernel and operating system development will be such as to provide a rather dynamic and oft-changing software environment. If the kernel and operating system development tasks are to be pursued in an efficient and expeditious manner, they must have access to a development facility without excessive regard for impact on production users. This task defines the requirement for the dedicated secure facility. (Planned)

Task 25 -- Audit and Surveillance Requirements Analysis: establishes the requirements for audit and surveillance techniques, both in a general ADP environment and for specific application to the secure general-purpose prototype. (Completed)

Task 26 -- Audit and Surveillance Design: Based on the requirements determined in Task 25, this task develops the design for a security audit subsystem for use with the secure general-purpose prototype system. Required kernel actions and appropriate audit strategies are defined. (In progress)

Task 27 -- Audit and Surveillance Implementation: The audit and surveillance subsystem designed by Task 26 is implemented to operate in the kernel and secure system environment. (Planned)

Task 28 -- Audit and Surveillance Integration: integrates audit and surveillance tools into the secure general-purpose prototype system. (Planned)

TECHNOLOGY TRANSFER

As we have seen, the prerequisite group develops technology and initial products for achievement of multilevel computer security. The secure general-purpose system development group applies the technology and develops a prototype of a multilevel secure "computer utility." The tasks of the technology transfer group, then, are the key to applying the results of the first two groups to meeting the specific computer security requirements of the community of Air Force computer users. These tasks produce specifications, usable products and engineering techniques in forms suitable for direct application by user commands and acquisition agencies. Specific sets of tasks in this group deal with demonstrating the utility of the kernel, providing support to ESD programs and the Air Force Data Service Center's multilevel secure Multics system, and specifying security requirements and controls for other Air Force systems.

Task 29 -- Brassboard Security Kernel Application Studies: The brassboard security kernel for the PDP-11/45 (or similar minicomputers) produces a secure (though small) computer system in an early time frame. A variety of proposed applications could benefit from the availability of such a secure computer. This task provides documentation and application guides for the brassboard kernel for direct use in operational systems. (Completed)

Task 30 -- Brassboard Kernel File System: The automated processing and correlation of data from tactical sensors requires concurrent processing of data at various classification levels. This task, first in a series that will produce a demonstration software system for securely processing data in a tactical environment, is directed toward design and implementation of a file system for the brassboard kernel. (Completed)

Task 31 -- Downgrading Mechanism Design and Implementation: A key requirement of the application discussed in Task 30 is the capability to selectively sanitize and downgrade sensor information. This task extends existing computer security technology and concepts to fit the downgrading requirement and procuces design and implementation of a downgrading mechanism for the brassboard kernel. (Completed)

Task 32 -- Demonstration Scenario Development and Demonstration: To substantiate the usefulness of the software system developed by Tasks 30 and 31, this task prepares a demonstration scenario for processing and downgrading information in a tactical environment. The scenario and demonstrations will illustrate situations and instances where the capabilities of the proposed system are necessary. (Completed)

Task 33 -- AFDSC Multics Security Evaluation: provided a preliminary evaluation of the suitability of the Honeywell Multics computer system for use in a multilevel (Secret-Top Secret) environment at Air Force Data Services Center. (Completed)

Task 34 -- AFDSC Multics Security Control: applies preliminary computer security modeling results to the specification, development, testing, and integration of security control enhancements intended to make Multics suitable for use in the two-level environment at AFDSC. The controls provide Multics with enhanced protection, and adapt it for use in a specific military security environment; however, they do not insure that the system can withstand malicious penetration efforts. (Completed)

Task 35 -- Follow-on AFDSC Multics Security Support: Once the AFDSC Multics Security controls are installed and operational, they must be subject to continued validation, review and enhancement. (A true security kernel would not require such a degree of continuing support, as it would be compact, isolated, and relatively stable.) This task provides the requisite support, and assists AFDSC in planning for eventual transition to the complete and secure systems developed by the tasks already described. (In progress)

Task 36 -- Jobstream Separator Requirements Analysis: investigated the application of a secure minicomputer to automation of the "color change" process at various WWMCCS sites. The jobstream separator offers a practical, immediate solution to the inefficiencies inherent in present security level change procedures. (Completed)

Task 37 -- Jobstream Separator Prototype: will design and implement a prototype jobstream separator for the Honeywell WWMCCS computers. Included in this task will be development of the security control minicomputer, suitable modification of the main computer's hardware and software and design of additional necessary hardware to permit automation of the "color change." (Planned)

Task 38 -- AFSC COMSEC Study: The requirements for a secure message system terminal to operate in a general office environment are studied. (In progress)

Task 39 -- AFSC COMSEC Design and Demonstration: extends the study done under task 38 by developing a prototype secure terminal to be used in conjunction with communications systems. (Planned)

Task 40 -- Secure Network Front-End Hardware Requirements Analysis: determines the hardware requirements for a secure network front-end processor in the WWMCCS environment and incorporates them into the design of the secure communications processor being developed in Tasks 11 to 16. (In progress)

Task 41 -- Secure Prototype Network Front-End Studies: The requirements and techniques for securely connecting WWMCCS computers into a Prototype WWMCCS Intercomputer Network (PWIN) are studied. In particular, the use of the secure front-end processor as a communications processor is examined. (Planned)

Task 42 -- TIPI Hardware/Software Studies: provides computer security support to the Tactical Information Processing and Interpretation (TIPI) Program Office. TIPI security requirements are analyzed and the operational impacts of security kernel technology are assessed. (Planned)

Task 43 -- AABNCP Requirements Analysis: analyzes the multilevel computer security requirements in the Advanced Airborne Command Post. (Planned)

Task 44 -- AABNCP Prototype Demonstration: provides a prototype verifiable computer system capable of providing the controlled data sharing required by the Advanced Airborne Command Post. (Planned)

Task 45 -- WWMCCS II Alternative Studies: The planning for a second generation of WWMCCS ADPE must begin early and include explicit provision for multilevel security. This task supports the WWMCCS II planning by establishing specific Air Force WWMCCS

II security requirements and by evaluating the alternative approaches to meeting WWMCCS II ADPE security requirements. (Planned)

Task 46 -- Follow-on WWMCCS II Support: continues the support initiated in Task 45 through specification, acquisition and evaluation of security elements of WWMCCS II ADPE. (Planned)

Task 47 -- Specification and Acquisition Guidance Documentation: The secure general-purpose system development group develops a verifiably secure "computer utility" system. While Air Force users can acquire secure computing capability by duplicating the prototype, it is vital that they also be able to specify a secure system for competitive acquisition from any of a variety of vendors. This task translates the prototype design and experience into sample secure system specifications and associated guidance for acquiring agencies. (Planned)

APPLICATION AIDS DEVELOPMENT

Certain subsystems, while not central to providing multilevel secure computer systems, will facilitate cost-effective use of secure systems in the field. The application aids development group produces a subsystem to facilitate data base management in a secure computer environment. This subsystem facilitates use of the secure system on a large data base of mixed classifications. The applications aid development group also produces an Air Force Computer Security Handbook.

Task 48 -- Secure DMS Model Development: If a data management system is to operate on files of several classifications simultaneously, and is to assure that a user accesses only a controlled subset of those files, the DMS must be based on a model which is compatible with the security kernel that controls it. This task provides a model on which such a data management system can be based. (In progress)

Task 49 -- Secure DMS Design: prepares a design for a secure data management system that implements the model developed by Task 48. (Planned)

Task 50 -- Secure DMS Implementation: implements a secure data management system as an application subsystem of the secure general-purpose prototype system. (Planned)

Task 51 -- Secure DMS Test and Evaluation: evaluates the operational utility of the secure data management system in the secure computer environment. (Planned)

Task 52 -- Design Handbook Task Definition: defines the requirements for and the contents of an AFSC Computer Security Design Handbook. The handbook would codify available information in order to guide designers of computer systems faced with security requirements. (In progress)

Task 53 -- Computer Security Design Handbook: develops the computer security handbook in accordance with the guidelines established in Task 52. The information is organized as a handbook suitable for periodic updating (Task 54). (Planned)

Task 54 -- Computer Security Design Handbook Maintenance: As development continues and new technologies become available, they must be transmitted to system designers. This task updates the computer security design handbook periodically (every six months to a year) to reflect new results, techniques and practices. (Planned)

SECURE COMPUTING ENVIRONMENT DEVELOPMENT

A secure multilevel computer system should extend the scope of the classified computing services provided to Air Force users. For example, individuals with small computing tasks to perform at the Secret level should be able to perform those tasks on a multi-user secure timesharing system. For computing service to be provided efficiently to users, it should be possible to place a terminal for Secret level processing in an office as one would a safe.

This group of tasks is aimed at developing more efficient terminal and communications security equipment for the interactive computing environment. While these developments are not necessary for multilevel computer security, they will provide for more cost-effective use of secure computer systems. (Planned)

A second set of tasks within this group provides rapid, safe means of rendering classified information on storage media inaccessible. This set is aimed specifically at the problems of processing classified information in the tactical environment and of making it possible to store or transmit media that contain classified information using ordinary containers.

Task 55 -- Secure Office Terminal Design and Implementation: develops a prototype of a secure terminal suitable for interactive computer applications, with integrated communications security equipment, for use in a general office environment (not a vault). This task builds extensively on experience gained in developing a secure terminal for use with communications systems.

Task 56 -- Integration of Secure Terminal and Multiplexed Cryptographic Equipment: integrates the secure terminal developed by Task 55 with the multiplexed cryptographic equipment developed by Task 58. (Planned)

Task 57 -- Secure Terminal Test and Evaluation: tests and evaluates the secure terminal for application with the secure prototype computer system. (Planned)

Task 58 -- Multiplexed Cryptographic Equipment Development: A secure front-end processor can control a single cryptographic device that provides security for a number of separate secure terminals, or for many users in a computer network. This configuration can reduce the cost, space and power required for cryptographic equipment at computer sites that serve numerous remote terminals. This task develops the required cryptographic equipment. (Planned)

Task 59 -- Integration of Multiplexed Cryptographic Equipment and Secure Front-End Processor: integrates the cryptographic equipment with the secure front-end processor. Application programs for the front-end processor will be needed to drive the multiplexed cryptographic device. (Planned)

Task 60 -- Multiplexed Cryptographic Equipment Test and Evaluation: provides operational test and evaluation of the multiplexed cryptographic equipment in an environment including secure central computer, front-end processor and secure terminals. (Planned)

Task 61 -- Emergency Denial Techniques Catalog: begins the development of techniques for emergency denial of access to classified information with a survey and catalog of potentially suitable techniques. This task will specifically consider application of media encryption techniques. (Planned)

Task 62 -- Emergency Denial Techniques Development: selects promising techniques from the catalog developed by Task 61 and develops prototype equipment for evaluation. (Planned)

Task 63 -- Emergency Denial Techniques Integration: For evaluation, the prototype denial techniques will be used with the prototype secure general-purpose system. This task integrates the denial prototype equipment into the secure general-purpose system. (Planned)

Task 64 -- Denial Techniques Test and Evaluation: assesses the reliability, effectiveness and compatibility of the prototype denial equipment. Not only must the equipment effect denial on demand, but it must also guarantee against accidental denial or loss of information. (Planned)

REFERENCES

1. J. M. Schacht, "Jobstream Separator System Design," ESD-TR-75-86, Volume I, The MITRE Corporation, Bedford, Massachusetts, May 1975.

2. J. P. Anderson, "Computer Security Technology Planning Study," ESD-TR-73-51, Volumes I-II, James P. Anderson & Co., Fort Washington, Pennsylvania, October 1972.

3. R. R. Schell, P. J. Downey, and G. J. Popek, "Preliminary Notes on the Design of Secure Military Computer Systems," MCI-73-1, Electronic Systems Division (AFSC), L. G. Hanscom Field, Bedford, Massachusetts, January 1973.

4. D. E. Bell and L. J. LaPadula, "Secure Computer Systems," ESD-TR-73-278, Volume I-III, The MITRE Corporation, Bedford, Massachusetts, November 1973 - June 1974.

5. D. E. Bell and L. J. LaPadula, "Computer Security Model: Unified Exposition and Multics Interpretation," ESD-TR-75-306, The MITRE Corporation, Bedford, Massachusetts, June 1975.

6. K. G. Walter, et al. "Initial Structured Specifications for an Uncompromisible Computer Security System," ESD-TR-75-82, Case Western Reserve University, Cleveland, Ohio, January 1974.

7. D. Bransted, "Privacy and Protection in Operating Systems," Computer, Volume 6, Number 1, January 1973, pp. 43-47.

8. W. R. Price, "Implications of a Virtual Memory Mechanism for Implementing Protection in a Family of Operating Systems," Ph.D. Thesis, Carnegie-Mellon University, Pittsburgh, Pennsylvania, June 1973.

9. E. L. Burke, "Synthesis of a Software Security System," MTP-154, The MITRE Corporation, Bedford, Massachusetts, August 1974.

10. D. E. Bell and E. L. Burke, "A Software Validation Technique for Certification, Part 1: The Methodology," ESD-TR-75-54, Volume I, The MITRE Corporation, Bedford, Massachusetts, April 1975.

11. J. K. Millen, "Security Kernel Validation in Practice," ESD-TR-75-54, Volume II, The MITRE Corporation, Bedford, Massachusetts, September 1975.

12. L. Robinson, P. G. Neumann, K. N, Levitt, and A. R. Saxena, "On Attaining Reliable Software for a Secure Operating System," 1975 International Conference on Reliable Software, Los Angeles, California, April 1975, pp. 267-284.

13. W. L. Schiller, "Design of a Security Kernel for the PDP-11/45," ESD-TR-73-294, The MITRE Corporation, Bedford, Massachusetts, December 1973.

14. E. W. Dijkstra, "The Structure of the 'THE' Multiprogramming System," Communications of the ACM, Volume 11, Number 5, May 1968, pp. 341-346.

15. W. L. Schiller, "The Design and Specification of a Security Kernel for the PDP-11/45," ESD-TR-75-69, The MITRE Corporation Bedford, Massachusetts, May 1975.

16. L. Smith, "Architectures for Secure Computer Systems," ESD-TR-75-51, The MITRE Corporation, Bedford, Massachusetts, April 1975.

17. J. C. Whitmore, A. Bensoussan, P. A. Green, A. M. Kobziar, and J. A. Stern, "Design for Multics Security Enhancements," ESD-TR-74-176, Honeywell Information Systems, 1974.

18. Steven B. Lipner, "Multics Security Evaluation: Results and Recommendations," ESD-TR-74-193, Vol. 1 (in preparation).

19. "Secure Communications Processor Architecture Study," ESD-TR-76-351, Honeywell Information Systems, 1976.

MISSION

OF THE

DIRECTORATE OF COMPUTER SYSTEMS ENGINEERING

The Directorate of Computer Systems Engineering provides ESD with technical services on matters involving computer technology to help ESD system development and acquisition offices exploit computer technology through engineering application to enhance Air Force systems and to develop guidance to minimize R&D and investment costs in the application of computer technology.

The Directorate of Computer Systems Engineering also supports AFSC to insure the transfer of computer technology and information throughout the Command, including maintaining an overview of all matters pertaining to the development, acquisition, and use of computer resources in systems in all Divisions, Centers and Laboratories and providing AFSC with a corporate memory for all problems/solutions and developing recommendations for RDT&E programs and changes in management policies to insure such problems do not reoccur.